```
*************************************************************
                                  ***
        /////////      ////   ////  ***  /////////
       IIIIIIIII/     IIII/  IIII/    ***IIIIIIII//
      IIIII/          IIII/  IIII/     IIII/  IIII/
      IIIII/          IIII/  IIII/     IIII/* IIII/
      IIIII/          IIII/  IIII/     IIII/***III/
      IIIII//////  // IIII///IIII/  // IIII///***I/  //
       IIIIIIIII/ II/  IIIIIIIII/  II/  IIIIIIII*** II/
                                                ***
*************************************************************
                  CIRCULO DE USUARIOS DE QL
*************************************************************
```

Fanzine mensual independiente para usuarios de Sinclair QL y compatibles

AÑO 2 NUMERO 15 DICIEMBRE DE 1989

Estamos en el año 9 Después de Sinclair. Toda Hispania está ocupada por las legiones de PCs. ¿Toda? ¡No! Un puñado de irreductibles QLs resiste todavía y siempre ante el primitivo invasor...

CONTENIDO

Con este número de CUQ se incluyen los programas siguientes:

- EDT_ESPERANTO
- TRAQZ_EXE
- LIFE_EXE
- ORPAN_BAS

Con este número de CUQ se incluyen las siguientes traducciones:

MANUAL GUIA DEL USUARIO DEL FOUNTED89 (CUQ15MAN)

Material preparado para próximos números:

Haber no es que tengamos mucho, pero si suficiente para preparar otro CUQ hoy mismo. Por lo pronto hay:

- Un programa experto en enfermedades
- Otro para calculos financieros
- Un acelerador de pantalla para THE SOLUTION
- Otra de BBS
- Comentario de una impresora
- Y si alguien envia algo más, muchisimo mejor (Gracias anticipadas).

INFORMACION SOBRE C.U.Q.

Compilación de colaboraciones y distribución: Salvador Merino
Títulos y texto "Información sobre C.U.Q." y sub-distribución: Marcos Cruz
Imagen de la portada: OTRA DE MIS CHAPUCILLAS CON XCHANGE (QUILL)

SE CONSIENTE LA REPRODUCCION TOTAL O PARCIAL DEL CONTENIDO DEL FANZINE, PARA USO CULTURAL Y NO COMERCIAL, POR CUALQUIER MEDIO FISICO, QUIMICO, OPTICO, MAGNETICO, SOLAR, MECANICO, HIDRAULICO, EOLICO, ELECTRICO, NUCLEAR O A PEDALES, ¡SIEMPRE Y CUANDO SE CITE LA PROCEDENCIA! (¿EH? ¡OJO!).

El fanzine C.U.Q. es un medio de comunicación e intercambio de ideas y conocimientos entre usuarios de ordenadores Sinclair QL y compatibles. El fanzine se realiza mediante la colaboración desinteresada de todas las personas que lo desean.

¿COMO SE PUEDE RECIBIR EL FANZINE?

El fanzine se envía a quien contribuye a su realización aportando alguna colaboración, o bien a quien lo solicita sin más.

Quienes sólo quieran recibir el fanzine, sin enviar ninguna colaboración para que sea incluida en él, deberán enviar UN disco formateado de 3.5 pulgadas o TRES microdrives formateados, junto con un sobre bien franquedo y con las señas escritas para el reenvío. Estos discos o microdrives SIN COLABORACIONES deben enviarse a: Marcos Cruz, Acacias 44, 28023 Madrid

¿COMO SE PUEDE COLABORAR EN LA REALIZACION DEL FANZINE?

Quien desee colaborar en la realización del fanzine, debe igualmente enviar UN disco de 3.5 pulgadas o TRES microdrives (aunque sólo uno contenga algo) junto con un sobre bien franqueado y con la dirección escrita para el reenvío. Estos discos o microdrives CON COLABORACIONES deben enviarse a: Salvador Merino, Ctra. Cádiz (Cerámicas Mary), 29640 FUENGIROLA (Torreblanca del Sol), MALAGA

¿Qué puede enviarse como colaboración? Cualquier cosa: artículos, cartas, opiniones, preguntas, respuestas, ideas, sugerencias... Si bien el tema central del fanzine es el ordenador Sinclair QL, y la informática en general, C.U.Q. acepta artículos sobre cualquier tema que pueda interesar a los lectores.

Con todas las colaboraciones recibidas a lo largo de un mes, se confecciona el fanzine correspondiente al mes siguiente. Los discos y microdrives con los boletines se echarán al correo en el último fin de semana del mes.

No deben enviarse más colaboraciones hasta haber recibido de vuelta los discos o microdrives con el fanzine del mes correspondiente.

FORMATO DE LAS COLABORACIONES

Los textos deben ser ficheros _doc del programa Quill, con los siguientes márgenes: izquierdo=1, derecho=80, sangrado=5. Los textos deben ir justificados a la derecha, y no deben emplearse caracteres no castellanos cuyo código ASCII sea mayor de 127. Tampoco deben usarse los corchetes cuadrados, ni las llaves, ni las barras vertical o inclinada hacia la izquierda.

Los artículos deben agruparse temáticamente según las siguientes secciones:

CAR	Cartas abiertas
NOT	Noticias
LIB	Comentarios de libros y otras publicaciones
PRO	Comentarios de programas
HAR	Comentarios sobre hardware, y montajes
BAS	Lenguaje SuperBASIC
FTH	Lenguaje Forth
LEC	Lenguaje C
ASM	Lenguaje ensamblador 68000
COB	Lenguaje COBOL
ZET	Z88
SPE	Spectrum
OFE	Ofertas, compra-venta, anuncios particulares
OTR	Otros temas de interés, aunque no sean informáticos.

Esta lista podrá ser ampliada o modificada por el recopilador, a sugerencia de los colaboradores.

En el caso de que la colaboración consista en preguntas, o respuestas, deberá igualmente incluirse en una de las secciones indicadas, según el tema de a que se refieran.

Todas los artículos que pertenezcan a una misma sección deben ir en un mismo fichero. El nombre de este fichero tendrá el formato siguiente: SSSxxnnn_doc,

donde "SSS" son los tres caracteres correspondientes a la sección de destino del texto, "xx" es el número de fanzine al que se envía el texto y "nnn" son las iniciales del autor.

En el texto, el título de cada artículo deberá ir en mayúsculas, en letra negrita, y a la izquierda de la línea (en el margen de sangrado).

En los artículos ha de evitarse incluir líneas en blanco, con el objeto de ahorrar espacio en las copias impresas.

Al final de cada cada artículo debe ir el nombre del autor, en la columna 40, seguido en la misma línea por la localidad y la fecha.

Algunas importantes recomendaciones a observar:

-Déjese siempre un espacio detrás de cada signo de puntuación (coma, punto, punto y coma, dos puntos) ¡pero no delante!

-No deben aislarse con espacios los signos de interrogación, admiración, guiones o paréntesis, sino que deben estar unidos a la palabra siguiente -si son signos de apertura- o anterior -sin son de cierre.

-Cuídense tanto la ortografía y las tildes como los errores de "tecleo". Para ello, es recomendable releer lo que se ha escrito.

¿EN QUE FORMATO SE RECIBE EL FANZINE?

El fanzine se recibe en forma de varios ficheros _doc del procesador Quill, al objeto de permitir el acceso al mismo de los usuarios que no tienen unidades de disco y/o ampliación de memoria.

El nombre de cada fichero indica el número de fanzine al que pertenece y la sección que contiene.

Para componer el fanzine debe cargarse el fichero CUQxxPOR (donde "xx" es el número de fanzine) y seguidamente deben UNIRse al final del texto, uno a uno, los ficheros de las restantes secciones en el orden que indique la columna "Sección" del índice del fanzine. Seguidamente puede alterarse el número de líneas por página, asignarse un pie de página y/o un encabezamiento y, si se desea, añadir cambios de página forzados para evitar que los títulos de secciones o artículos queden al final de alguna página. Finalmente, se ha de grabar el fanzine completo en disco y/o sacar una copia por impresora.

Si, por falta de memoria, no puede componerse todo el fanzine entero, deberá imprimirse por partes, a juicio de cada uno.

Una vez tenemos el fanzine en papel, habrá que rellenar a mano los números de página del índice.

¿PUEDEN INCLUIRSE DIBUJOS EN EL FANZINE?

Normalmente la portada llevará un dibujo (preferiblemente relacionado con alguno de los artículos del fanzine). El dibujo será un fichero de pantalla, en modo 4 u 8. El nombre de estos ficheros debe ser: DBPxxnnn_MODEm, donde "m" es el modo de pantalla en el que están realizados. Como en los textos, "xx" es el número del fanzine de destino, y "nnn" son las iniciales del autor.

El dibujo elegido por el recopilador para ser la portada del mes, se incluirá en el fanzine con el nombre CUQxxDBP_MODEm (donde "xx" es el número de fanzine y "m" indica el modo de pantalla, 4 u 8). Este fichero deberá emplearse para sacar un "hard-copy" sobre el folio de la portada.

La inclusión de dibujos en los artículos aún no es posible, pero están en estudio varios métodos posibles.

¿COMO ENVIAR PROGRAMAS AL FANZINE?

El fanzine C.U.Q. sirve también como intercambio de programas (PROGRAMAS REALIZADOS POR LOS MISMOS COLABORADORES, NO PROGRAMAS COMERCIALES). Estos programas pueden estar realizados en cualquier lenguaje disponible para el QL. En la medida de lo posible, y siempre que su longitud no sea excesiva, los textos fuente de dichos programas deben incluirse en un artículo (en la sección del lenguaje correspondiente) en el que se explique el funcionamiento u otros detalles que el autor considere oportuno. Los listados de programas que se incluyan en el boletín, no deberán tener líneas con más de 80 caracteres. Si el autor prefiere no ceder al fanzine el texto fuente, deberá igualmente escribir las instrucciones de manejo en un pequeño artículo, para su inclusión en el fanzine.

Normas a tener en cuenta para enviar programas al fanzine.

-Póngase a cada fichero una extensión adecuada que indique su tipo.

-Póngase a todos los ficheros de un mismo programa un mismo nombre inicial, que no coincida con alguna de las referencias de los programas ya existentes. Si entre los ficheros de un programa existe un fichero _doc del Quill, o de otro paquete de Psion, el nombre inicial común a todos los ficheros no podrá, por tanto, ser más largo de 8 caracteres.

-No deben emplearse en los nombres de los ficheros caracteres que no sean

aceptados directamente, es decir, que no sea necesario encerrar el nombre entre comillas para manipular el fichero.

-Deben escribirse instrucciones detalladas del manejo de los programas, bien dentro del mismo programa, o en un artículo que aparezca en el fanzine, o bien en un fichero _doc.

Los programas deben enviarse, como las colaboraciones escritas, a: Salvador Merino, Carretera de Cádiz (Cerámicas Mary), 29640 FUENGIROLA (Torreblanca del Sol), Málaga.

EDITORIAL

-Hoy el PSOE ha renovado su titulo por tercera vez en este siglo (yo confieso no haberlos votado).

La portada ha sido dedicada a nuestro brillante y siempre a nuestro servicio QUILL (nuestro procesador de textos de toda la vida). Creo que después de seis años, se lo merece.

Pasando revista a la actualidad (ver sección noticias), vemos que estamos todavía vivitos y coleando (buen regalo de Navidad, ¡Nooo...!).

Para aquellos que querian un cursillo de Assembler 68000, os deseo un apetitoso desayuno con la sección de este mes. Espero haber sido útil (¡Pobrecillo el QL programación Avanzada y QDOS COMPANION con hoja para alli, hoja para aca!).

Si nos fijamos en la competencia, vemos que Atari, después de tres años sin sacar una novedad en la gama ST, ha lanzado este verano dos novedades en Alemania, el portable STacey y el Atari TT basado en el Motorola 68030.

Si bien el Transformer de Ultrasoft que iba a ser comercializado por CHON no se ha dejado ver, en el mundo ST ha salido un nuevo emulador PC llamado PC-speed, que consiste en una tarjeta hardware que contiene un 8086 (el NEC V30) y ofrece 4 veces la velocidad de un IBM PC (la potencia de un AT). En España se puede encontrar por 38.000 ptas + IVA.

Lo único en común que poseen THE NEXT, MAC IIx y Atari TT, es la palabra UNIX.

Aunque nuestro CIRCULO solamente está interesado en el Motorola 680XX y nuestro, siempre admirado, Tio Clive, os he actualizado de alguna manera el mundo ATARI. La razón es muy sencilla. Algún dia, tarde o temprano, voy a tener que actualizar o sustituir mi QL por uno basado en el Motorola 68030 que pueda correr un sistema operativo compatible QDOS. Y según Roy Barber (el actual editor de Quanta) ya ha visto el nuevo "QL type operating system running on the ATARI ST" por Tony Tebby en uno de sus recientes mitines corriendo el software del QL 4 o 5 veces más rápido (el doble que el emulador actual). Naturalmente, de aqui al 92 puede pasar de todo.

Apostando por el futuro de la familia Motorola 680XX en los años 90, os dejo esperando hasta el próximo año.

¡Feliz Navidad!

S. Merino

NOTICIAS

UN OTOÑO'89 MUY MOVIDO

- THOR INTERNACIONAL estaba esperando en septiembre la licencia para exportar un pedido hecho por escrito a cinco empresas en Rusia, que consiste en 1.000 ordenadores y periféricos por valor de dos millones de libras. También están negociando con Unisoft una licencia para transportar una versión del sistema operativo UNIX al THOR 16/20 (el desarrollo podria costar sobre trescientas mil libras).

- El QL emulator on the Atari ST ya no es comercializado por Strong Computer System desde principios del pasado verano. ¡Qué no cunda todavía el pánico!.

JOCHEN MERZ SOFTWARE comercializa en Alemania el AQMULATOR que consiste en

una nueva versión del anterior emulador que emula QSound, el más nuevo de los QJUMP Drivers, Supertoolkit II, Menu-driven port control, Hotkey System II, MODE 4 & MODE 8, y algo más por solamente 155 libras (agregar 2 libras para P&P). JOCHEN MERZ SOFTWARE, Im Stillen Winkel 12, 4100 Duisburg 11, W-GERMANY.

- Miracle Systems comercializa su QL HARD DISK con 40 Mbyte por 449 libras (export 405 libras).

-ABC ELECTRONIC comercializa una nueva versión de la MEGA RAM compatible con la Trump Card por 195 libras. También Interface disco duro por 129 libras y QIMI por 39 libras sin ratón y 62 libras con ratón. Keyboard Interface por 52 libras. ABC Elektronic, Hügelstr. 10-12, 4800 Bielefeld 1, W-GERMANY.

- Digital Precision ha lanzado en octubre el PC CONQUEROR que consiste en una nueva versión del SOLUTION, pero un 80 % más rápido y más compatible PC (Dicen que no conocen ningún programa comercial que no pueda correr con el nuevo emulador). Ahora en formatear un disco de 720K tarda 123 segundos y medio minuto en arrancar el MS-DOS.

- MINERVA THE ROM consiste en una nueva ROM sin más de 100 BUGs conocidos en las antiguas ROMs AH a MG. Ha sido escrita por Qview y THOR INTERNATIONAL, y han tenido mucho cuidado para evitar infringir el Copyright (de Amstrad).

- Los cartuchos de MDV vuelven a fabricarse, programas nuevos para Navidades, la subscripción por 12 a la revista SINCLAIR QL WORLD baja a 24.70 libras para Europa, etc...

S. Merino, Fuengirola, 30/10/1989

CARTAS ABIERTAS

NOTICIAS DE PAMPLONA

Hola Salvador,

No te mando MDVs porque no he hecho nada este verano.

Estoy con el fin de carrera y lo acabaré en octubre si todo sale bien.

He visto y tengo en mi poder todos los números del CUQ. Me los pasa Lacasa. Asi no te lias con copias innecesarias.

También me ha pasado el toolkit II y Qliberator traducido. Se me ha abierto el cielo. Ahora podré programar a gusto.

He leido que hay más traducciones en curso. Deberias anunciar cuales son, porque yo estoy pasando el BASIC-ALLY con sangre y sudor, y si está traducido me da un sincope.

Los CUQs son muy buenos y tienen un tono desenfadado muy ameno. La verdad es que los que no sabemos aprendemos hasta de los chistes.

Quizá (y sobre todo ahora que no llega el QL WORLD) sería interesante insertar la publicidad de articulos con las direcciones (total, el que piense que se repite mucho, con no escribir la hoja listo).

También están muy bien los articulos sobre la evolución de los micros,... los cursos de toolkit II ... No os corteis pensando que ya lo sabemos, porque no. Y en las revistas (menos Microhobby) vienen más que anuncios y chorradas.

Podeis hacer cursos de C. Seguir los de C.M. ....

Respecto a mi PC. Todavia no lo he comprado. Cada vez que voy a la tienda, bajan de precio. Ya me habia decidido por un AT SHINEIDER de 249.000 ptas disco 3 1/2 720K W 20 Megas.... Y ahora me dicen que los Amstrad 386 bajan 300.000 ptas de golpe. Me va a dar un ataque.

Espero que a parte de programas de arquitectura haya buenos juegos y pueda disfrutar de las teclas aunque sea.

En cuanto tenga algo te lo mando.

En un próximo futuro tengo intención de comprar la Trump Card.

Un saludo:

Fernando Martinez, Pamplona, 22/9/1989

UN QLMANICO MAS

No se hasta que punto puede servirle a las gentes pero de todas formas ahi va. La mando tal como la he recibido y se trata de una subcrición por un año a QL WORLD. la forma es la siguiente:

Se pide en un banco un talón en libras para enviar a Londres por el importe

de 24,70, y junto con la dirección se manda, al cabo de dos meses recibes durante un año la revista además sin problemas de ningún tipo.

Os cuento este sistema, ya que hasta haberlo conseguido yo no sabia la manera de hacerlo si a alguien le interesa ahi lo tiene.

el hecho de ponerlo en ingles es por lo que sigue: (no tengo ni idea de traducirlo) (no se lo digai a nadie) el ingles mio es de aprobar con 5 raspado y una buena dosis de voluntad del profesor.

Lo primero es lo primero, me presento.

UN QLMANICO MAS.

Me llamo Javier Gonzalez Santana

Vivo en Sevilla Ciudad Verde 126, Dtp. 41007.

Ahi teneis a vuestra dispocisión, sin tiempo, pero un forofo del QL. Yo para dejar desbancado a las gentes de PCs les digo si me dices donde está el PC que haga la mitad de cosas a la vez que mi QL te doy por el lo que pidas, nadie acepta la apuesta.

Esta es la carta que me envian, lo más literal posible.

FOCUS MAGAZINES. Ltd.

SUBSCRIPTIONS Deptt, P.O. Box No. 74.Tonbrige, Kent TN126DW

Priority Subscription Renewal Coupon.

Dear Reader,

Your magazine Subcription will soon expire and to ensure that supply of the publication is not interrupted, I have tanken the opportunity to send you this priority renewual coupon.

As a subscriber, you will know convenient it is to have each issue of the magazine sent to your home or office directly upon publication. A esspedy return of the attached coupon will guarantee that this fast, reliable and disappointment-free service will continue without pause.

The year ahead is set to be a particulary interesting and exciting one for both the magazine and its readers - I look forward to receiving your renewal so that you may continue to be part fo it.

Please detach the completed coupon and return it to me today unlees, of course, you have alreday renewed your subscription recently.

Assuring you of my best attention,

Publisher.

P.S. As well as being the dest way to guarantee that you never miss this informative and entertainig magazine, an inexpensive regular subscripction will allow you to take advantage of the subscribers only affers planned for the coming moonths Dont be left out.

El Destinatario al que yo mandaré el talón en libra será como el anterior es decir

SINCLAIR QL WORLD, GREENCOAT HOUSE,

FRANCIS STREET, LONDON SW1P 1DG.

Con estas señas y el talón en libras, las revistas ha ido llegando con toda puntualidad.

Sin más, espero que sirba.

Vuestro amigo Javier.

COMENTARIOS SOBRE HARDWARE

PRODUCTO: QL 5.25" DISC DRIVE
DISTRIBUIDOR: MIRACLE SYSTEMS
25 BROUGHTON WAY
OSBALDWICK
YORK YO1 3BG (UK)
PRECIO: 114 LIBRAS (puesto en España)

Hace tiempo que estaba pensando en la posibilidad de ampliar mi QL con una unidad de disco de 5.25" más que nada para poder leer los discos de los PCs (y otros ordenadores) directamente ya que la mayoria usan 5.25" 360K.

He enviado las libras el 30 de septiembre y he recibido el paquete el 4 de

octubre. El paquete era gigante y he tenido que pagar 110 ptas en correos (no sé de qué). Las caracteristicas son:

La unidad de disco tiene unas medidas de 32x15x7 cm.

Usa diskettes de 5 1/4", 48tpi, 40 track, DOUBLE-SIDED, DOUBLE-DENSITY dando 360 K de capacidad.

Hay que comprar una clavija ya que vienen los 3 cables pelados.

No hay interruptor de encendido.

En la parte posterior existe un duplicado del conector del interface de disco.

En la cara frontal hay un interruptor con un 1 y un 3 a cada lado. Sirve para convertir la unidad de 5.25" en flp1_ o flp3_, y la unidad de 3.5" original flp1_ será siempre la inversa de 5.25".

Cuando me decidi a probar la unidad de disco 5.25" me encontre con el problema de que no funcionaba. Rapidamente me decidi a abrir la caja, pero desafortunadamente en vez de tornillos aquello estaba atornillado o grapado con unos tapones de plástico negro. En el intento de destaponar aquello, los tapones quedaron seriamente destruidos.

Los tapones negros deben tratarse de un sistema de seguridad (o economia) para detectar si una unidad ha sido abierta, pues ello invalida la garantia. Consiste en dos plásticos: el que se introduce primero es un tapón hueco que cuando se introduce el segundo, que tiene forma de chincheta, en él, se habre haciendo la fuerza de unión que deberia hacer un tornillo. Creo que la forma más fácil de abrirlo es con un objeto punzante perforando el tapón en el mismisimo centro y haciendo que la pua de la chincheta sea empujada al interior de la carcasa.

Una vez abierto la caja, ví con alegria el motivo por el cual aquello no queria trabajar. La clavija del cable plano no estaba enchufada al conector plano de la tarjeta de disco.

Segundo problema, esta vez de la Trump Card, sabia que mi controlador solamente podia direccionar un máximo de 2 unidades de disco, pero ..... (las últimas Trump Card creo que pueden controlar 4 unidades). Lo que significaba que la única manera de usar las dos, como estaban, eran como flp1_ usando el conmutador.

Bien, mi unidad de 3.5" está configurada como flp1_. La unidad de 5.25" como flp3_. El invento del interruptor consiste en un puente hecho en el mismisimo cable plano. Nada me impide configurar la unidad de 5.25" como flp2_. Esto último se hace cambiando un plástico verde de posición en unas puas que hay cerca del conector plano de la tarjeta de disco (están enumeradas como 0,1,2 y 3).

Dicen malas lenguas que las unidades de 5.25" son muy ruidosas, pero ésta con la marca TEAC se lleva todos los premios, porque produce un "Taca-taca-taca..." muy molesto cada vez que se usa cualquiera de las dos unidades de disco y cuando se introduce o se saca el disco de 5.25". La mayoria de las impresoras hacen menos ruido que esta unidad de disco de 5.25", y menos mal que puedo desconectarla y conectarla cuando lo desee.

Si hay algo increible es que el cable plano sumando todos los trozos tiene bastante más de 2 metros, que es el máximo recomendado para cuando se tiene 4 unidades de disco conectadas a la vez. Hasta ahora he podido comprobar que todo funciona correctamente a pesar de la longitud del cable si se tienen las dos unidades de disco conectadas a la red eléctrica a la vez (cada una posee su propia fuente de alimentación).

El interruptor no sirve para nada con el QL 5.25" DISC DRIVE configurado como flp2_, pero siempre se puede cambiar el puente para haga lo mismo con flp1_ y flp2_ pudiendo reconfigurar las dos unidades de disco con solamente pulsar el interruptor. Solamente se necesita una cuchilla de afeitar, un soldador, estaño y saber lo que estamos haciendo (podemos cortarnos o quemarnos un dedo sin contar el poder equivocarse de cables).

Los discos de 5.25" me cuestan a 80 ptas la unidad frente a las 250 ptas de 3.5" (aunque este verano he podido comprarlos a 190 ptas). La diferencia de precio en el Kbyte no es mucha que digamos.

Tener unidad de 3.5" y 5.25" puede tratarse más de un capricho que de una necesidad, porque los discos de 5.25" tienen pegas por todos lados (transporte dificil, se introduce el polvo por doquier,....) y están retirandose del mercado (aunque el parque instalado es enorme).

Parecia que no se podia escribir mucho sobre una unidad de disco, pero ya veis.

S. Merino, Fuengirola, 5/10/1989.

COMENTARIOS DE PROGRAMAS

D-LAR

Para aquellos que ya hayan probado los graficos del Mejillones les dire que este es otro programa de Indiana Jones, en busca del cartucho perdido. Eso si, retocado convenientemente para aprovechar las facilidades del -multitarea.

La idea clave para comprender el programa es imaginar que -estamos utilizando una pecera en lugar de un monitor y que podemos ver como estan evolucionando la infinidad de celulas que salen del agua del grifo.

Pero esto solo es teoria, los impacientes ya habran podido comprobar que D-LAR no es mas que una version del famoso juego de ordenador LIFE. Dicho programa lo desarrollo originariamente John Conway durante su estancia en el Gonville and Caius Collge de Cambridge, en Gran Bretaña; aunque fue Martin Gardiner quien divulgo este juego en el mundo entero a raiz de la publicacion de un articulo suyo sobre el tema el el numero de octubre de 1970 de la revista Scientific American (Investigacion y Ciencia).

En LIFE las celulas nacen, se reproducen y mueren de acuerdo con las reglas inventadas por Conway. Una celula en una red (se imagina que la colonia de celulas se desarrolla en forma de red) esta rodeada por ocho mas y el estado de esas ocho dicta lo que pasara con la celula en cuestion en la siguiente generacion.

Las citadas reglas son:

- Si a una celula la rodean dos o tres mas, sobrevivira en la siguiente generacion.
- Si hay tres, y solo tres, celulas llenas junto a un espacio vacio, nacera una nueva celula en dicho espacio.
- Una celula que tenga cuatro celulas vecinas vivas morira en la siguiente genaracion.

Aunque en principio pueda pensarse que son unas reglas muy sencillas, en la practica producen dibujos que son mucho mas atractivos y menos predecibles de lo que uno se imagina tras leer las reglas.

El programa parte de una colonia definada por el usuario y la desarrolla una generacion. A partir de ahi las cosas son sencillas: mirar la pantalla y actuar en consecuencia. Solo hay una cosa que no aparece en los menus, el redibujado de pantalla; podemos redibujar toda la colonia en cualquier momento pulsando la tecla F4. Esta opcion de redibujado es la que permite tener varios programas en la memoria cada uno con su colonia e ir observando cada una de ellas en su evolucion.

Sergio Montoro. Madrid, octubre 1989.

VERSION DE "THE EDITOR" PARA TEXTOS EN ESPERANTO

Para emplear un tratamiento de textos en un idioma que posee caracteres que no están incluidos en el juego de caracteres estándar del ordenador, hacen falta tres cosas:

1.-Modificar el juego de caracteres, sustituyendo algunos de los ya existentes, según un cierto criterio, por los que sean necesarios.

2.-Facilitar el acceso por teclado de los nuevos caracteres.

3.-Hacer la traducción de dichos caracteres por los códigos adecuados de la impresora, antes de imprimir.

La potencia y flexibilidad de The Editor permite hacer todo eso con gran facilidad en el caso que nos ocupa. El objetivo es modificar The Editor para escribir textos en Esperanto. El Esperanto posee varios caracteres con supersigno, que no están incluidos en el juego de caracteres del QL, así que habrá que seguir los tres puntos citados:

1.-Caracteres:

Para modificar el juego de caracteres de The Editor basta emplear un diseñador de caracteres normal, como el que viene con Turbo. No obstante, lo primero que hay que hacer es determinar qué códigos vamos a emplear para los nuevos caracteres. En nuestro caso hemos empleado los códigos 128, 129, 130, 132, 133 y 134 para las nuevas minúsculas, y 160, 161, 162, 164, 165 y 166 para las nuevas mayúsculas. La razón es que a partir del carácter 128 se encuentran en el QL los caracteres internacionales en minúsculas, y a partir del 160 en mayúsculas. Así pues, comenzando desde ambos códigos, se han empleado aquellos que no están ocupados por un carácter del castellano, como es el caso de la "é",

que ocupa el código 131, y su correspondiente mayúscula que ocupa el 163.

Para que los caracteres con supersigno no quedasen más bajos que sus equivalentes sin supersigno, se ha empleado un juego de caracteres de menor altura, no el original de The Editor.

2.-Teclado:

Para facilitar el acceso a los nuevos caracteres, se ha recurrido a comando "ALTKEY" del Toolkit II. Con él, logramos acceder a los caracteres con supersigno simplemente con la tecla "ALT" y el carácter correspondiente. De este modo no es necesario etiquetar el teclado para indicar la posición de los nuevos caracteres. Por ejemplo, la "C" con supersigno se obtiene con la pulsación de "ALT" y "C".

3.-Impresora:

La traducción de los nuevos caracteres para su impresión puede lograrse mediante programas especiales destinados a este fin, pero en The Editor tenemos otra posibilidad, algo menos "transparente" para el usuario pero muy económica y sencilla: crear un fichero de comandos de The Editor que realice la traducción antes de la impresión. Este fichero, que ha de ser llamado con el comando "RC" una vez el texto a imprimir está cargado, tiene de nombre "edt_esperanto_ord", y puede ser modificado a voluntad si el tipo de impresora lo requiere, aunque las traducciones que realiza son las más simples posibles y funcionarán en cualquier impresora, pues no se emplean caracteres gráficos.

Esto ha sido un ejemplo aplicado a The Editor, pero ha pretendido ser ilustrativo del procedimiento a seguir en casos similares con cualquier idioma y con cualquier tratamiento de textos. Algún programa no se podrá adaptar, como es el caso de Quill, al menos en lo relativo al juego de caracteres (pues el teclado se solucionaría igual, con "ALTKEY", y la impresora mediante un programa driver, puesto que las 10 traducciones que permite hacer Quill no son suficientes), ¿o alguien sabe cómo cambiar el juego de caracteres de Quill?

Bueno, en este número de CUQ adjunto copia completa de The Editor lista para funcionar en Esperanto, ¡a ver si os animáis a aprender la Lengua Internacional!

Marcos Cruz, Madrid, 9/10/1989

LENGUAJE SuperBASIC

ORPAN_BAS

Te envio un programa en basic que permite calcular la orientacion de los paneles de energia solar. Espero complementarlo con otros relacionados que nos permita calcular la cantidad de paneles solares necesarios y sus anexos.

LUIS JOSE GARCIA GARCIA
VAZQUEZ AROCA 14 4-1
14005 CORDOBA

BBS

NORMAS CCITT

Siempre que de alguna manera accedemos al mundo de las comunicaciones o leémos algún artículo de telemática o llega a nuestras manos algún folleto de propaganda donde vienen las especificaciones técnicas de un modem,por ejemplo,vemos que comienzan a aparecer continuas referencias a normas CCITT que para muchos son ,en sí,mas un lío que una ayuda.

Si leemos que un modem cumple las normas V21, V22 y via de acceso V24,el articulista o propagandista presupone que nosotros estamos perfectamente enterados del significado de éstos términos, y me da la personal impresión que

ésto deja muy mucho de ser verdad.

Como puede ser de ayuda para alguno de vosotros,voy a continuación y de una manera sucinta a intentar describir el significado,si no de todas,al menos de las mas utilizadas a nuestra escala.

El CCITT(Comite Consultivo Internacional de Telefonia y Telegrafia,mas o menos)es un organismo internacional que entre otras cosas intenta poner orden dentro del mundo de la telematica que por ser tan reciente y estar en continuo cámbio hace que ante cada aparición de una técnica nueva,las administraciones nacionales intenten imponer cada una de ellas sus propias normas.

Aunque sus resoluciones solo tienen caracter consultivo,que no ejecutivo,la verdad es que se inponen de facto porque nadie quiere quedarse aislado ante la aceptación de dichas normas por los demas paises.

A continuación os detallo aunque sea sucintamente cada una de las normas más utilizadas.

Sólo me voy a referir a aquellas que utilizan como medio de transmisión la R.T.C (Red Telefónica Conmutada) o sea aquellas cuyo soporte son los dos hilos de un enlace telefónico y cuyo acceso al terminal distante es esporádico y siempre a través de una central telefónica automática.

Cuando el enlace es permanente por una línea dedicada en exclusiva se llaman enlaces "punto a punto".

CCITT V21.-

Es el mas lento de todos ellos ya que su velocidad es de 300 baudios y en modo asíncrono (no existe control de sincronismo entre ambos terminales sino que el comienxo y fin de cada caracter se determina por impulsos de arranque y parada incluidos dentro de la misma codificación.).

El tipo de transmisión es FULL DUPLEX que quiere decir que se mantiene simultaneamente ambas direcciones activas,ejemplo :una conversación telefonica;a base de utilizar diferentes frecuencias vocales para la Tx y Rx de cada dirección.

Cómo la frecuencia de Tx(Transmisión) de un terminal debe de coincidir necesariamente con la de Rx (Recepción) del distante,en V21 hay dos estados que se pueden elegir arbitrariamente que son ORIGINATE MODE (MODEM QUE LLAMA) y ANSWER MODE (MODEN LLAMADO).

Es lógico pensar que,como norma,el terminal que genera la llamada telefonica se pone en modo LLAMADA y aquel que espera dicha llamada (bien en respuesta manual o automatica)estará siempre en modo RESPUESTA.

Esta norma V21 es la equivalente ,aunque con frecuencias distintas del standard americano conocido como BELL 103.

página 1

Las frecuencias estandarizadas para esta norma son:

Norma	Frecuencias en Herzios			
	Emisión		Recepción	
	0	1	0	1
V21 LLamada	1.180	980	1.850	1.650
V21 Respuesta	1.850	1.650	1.180	950

CCITT V22

Se trata de una mejora indudable de la norma anterior ya que con sus mismas caracteristicas de asincronismo y full duplex,la velocidad se eleva a 1200 baudios y la tarifa telefónica en consecuencia se reduce.

En esta norma se puede trabajar tambien en 600 baudios si las condiciones de la linea asi lo impusieran (ruidos,microcortes,etc).

CCITT V23

En ésta norma se trabaja a 600/1200 baudios,asincrono,half duplex (semiduplex o sea en ambas direcciones pero no simultaneamente,ej:un walkie-talkie).

Si la comunicación se realiza en ambas direcciones en 600 baudios entonces se conoce como MODE 1.Si es en 1200 baudios se llama MODE 2.

Cuando el servicio llamado es un VIDEOTEX o servicio semejante en el que la mayor parte de la comunicación se realiza en un solo sentido entonces se suele

utilizar un canal añadido para control de la comunicacion que funciona a 75 baudios y que se llama canal de retorno.
Esta norma V23 es la equivalente ,aunque con frecuencias distintas,del standar americano conocido como BELL 202.

CCITT V26

Es la norma de trabajo a 1200/2400/4800 baudios,en modo síncrono (Uno de los terminales(Maestro) manda constantemente impulsos de reloj que hacen funcionar al colateral(Esclavo) sincronizado con él.
Si el modem trabaja a 1200/2400 se le conoce como tipo V26 BIS y si trabaja a 2400/4800 se le llama V26 TER.

CCITT (otras)

La V24 no es una norma de trabajo sino aquella que determina las caracteristicas de funcionamiento,tensiones,cableado,etc de la superfamosa puerta serie RS 232 que por su importancia merece un proximo trabajo aparte.
Asi mismo la V25 describe las caracteristicas que ha de cumplir el sistema auxiliar del modem para las modalidades de llamada y repuesta tanto manual o automatica.
Al mismo tiempo determina el modo de control del TD sobre el modem,conjuntos de comandos para entendimiento entre ambos,etc.
Junto con la V24 la analizaremos el próximo més.

página 2

Un saludo a todos y animarse en colaborar,que no es tan dificil.

Antonio Rodriguez Hdz.
Aptdo 2107.
30000.-MURCIA

PD)
Como siempre que os doy mis datos lo hago con al apartado de correos que es donde me resulta comodo,no os he dado jamas mi dirección que es la siguente por si os coincide el poder hacerme alguna visita.

Antonio Rodriguez Hernandez
C/San Francisco,3 (Tfno 968-802157)
Javali Nuevo (MURCIA)

ASSEMBLER

TRADUCTOR QL/Z88

Durante este pasado verano me he entretenido en escribir o, mejor dicho, traducir la versión del programita Traductor QL/Z88 en Superbasic a Assembler 68000 bajo QDOS. Escribirla en mi Z88 en sucio (normalmente solia hacerlo en papel en el pasado) ha sido fácil, pero ensamblarla en el QL con el ensamblador de Metacomco es otro cantar, pues normalmente mis programas en assembler nunca corren bien a la primera. Pero después de unos veintitantos ensamblados ya tengo una primera versión cero errores y totalmente depurada.
El programa ha sido escrito con la idea de que sirva de ejemplo didáctico a todo aquel que esté interesado por el assembler, pero no sabe, no pregunta.
Es una traducción casi exacta a la versión Turbo-compilada con la excepción de que al traducir un fichero QL tipo EXP del Turbo_Xchange se elimina un carácter de fin de linea que sobra (recordad que el Xchange funciona con un pequeño emulador MS-DOS y los PCs usan dos caracteres para el fin de linea).
Uno de los detalles más preciosos, que el lector va a ver observando los comentarios, son las traducciones de los comandos Superbasic traducidas directamente a su equivalente llamada al QDOS.

Como adorno, vamos a tener en cada llamada actualizados a D3, que contiene el Timeout, y A0, que tiene el ID (aunque sea el mismo valor que contenian). Estos dos últimos normalmente después de una llamada al QDOS no son alterados, lo que significa que solamente hay que modificar sus valores cuando deseemos cambiar de ID o Timeout.

```
    * Empezamos abriendo los canales
           moveq.l     #-1,d1
           moveq.l     #2,d3
           lea         con3,a0
           moveq       #$1,d0     io.open
           trap        #2
           lea         id3,a1
           move.l      a0,(a1)
           moveq.l     #-1,d1
           moveq.l     #2,d3
           lea         scr4,a0
           moveq       #$1,d0       io.open
           trap        #2
           lea         id4,a1
           move.l      a0,(a1)
           moveq.w     #1,d1       csize#4,1,0
           moveq.w     #0,d2
           moveq.w     #-1,d3
           move.l      id4,a0
           moveq       #$2d,d0
           trap        #3
           moveq       #0,d1        paper#3,0
           moveq.w     #-1,d3
           move.l      id3,a0       sd.setpa
           trap        #3
           moveq       #1,d1        border#3,1,255
           move.w      #255,d2
           moveq.w     #-1,d3
           move.l      id3,a0
           moveq       #$c,d0       sd.bordr
           trap        #3
bucle
           move.w      #-1,d3       cls#3
           move.l      id3,a0
           moveq       #$20,d0      sd.clear
           trap        #3
           moveq.w     #1,d1        over#3,1
           moveq.w     #-1,d3
           move.l      id3,a0
           moveq       #$2c,d0      sd.setmd
           trap        #3
           moveq.w     #2,d1        csize#3,2,1
           moveq.w     #1,d2
           moveq.w     #-1,d3
           move.l      id3,a0
           moveq       #$2d,d0      sd.setsz
           trap        #3
    * Aqui podeis ver un ejemplo de bucle FOR-NEXT
           moveq.l     #7,d4
           moveq       #0,d5
traductor
           move.w      d5,d1
           moveq       #-1,d3
           move.l      id3,a0
           moveq       #$29,d0     sd.setin
           trap        #3
           move.w      #138,d1     cursor#3,138-i,30-i
           sub.w       d5,d1
           move.w      #30,d2
           sub.w       d5,d2
           moveq.w     #-1,d3
           move.l      id3,a0
           moveq       #$17,d0      sd.pixp
           trap        #3
```

```
moveq.w     #9,d2       print#3,"TRADUCTOR"
moveq.w     #-1,d3
move.l      id3,a0
lea         mensaje,a1
moveq       #$7,d0
trap        #3
addi.w      #1,d5
dbf         d4,traductor
moveq.w     #0,d1       over#3,0
moveq.w     #-1,d3
move.l      id3,a0
moveq       #$2c,d0      sd.setmd
trap        #3
moveq.w     #2,d1       csize#3,2,0
moveq.w     #0,d2
moveq.w     #-1,d3
move.l      id3,a0
moveq       #$2d,d0     sd.setsz
trap        #3
moveq       #2,d1       ink#3,2
moveq.w     #-1,d3
move.l      id3,a0
moveq       #$29,d0     sd.setin
trap        #3
moveq.w     #9,d1       at#3,7,9
moveq.w     #7,d2
moveq.w     #-1,d3
move.l      id3,a0
moveq       #$10,d0     sd.pos
trap        #3
moveq.w     #14,d2      print#3,"1   QL  >  Z88"
moveq.w     #-1,d3
move.l      id3,a0
lea         mensaje1,a1
moveq       #$7,d0      io.sstrg
trap        #3
moveq.w     #9,d1       at#3,9,9
moveq.w     #9,d2
moveq.w     #-1,d3
move.l      id3,a0
moveq       #$10,d0     sd.pos
trap        #3
move.w      #14,d2
move.w      #-1,d3
move.l      id3,a0
lea         mensaje2,a1
moveq       #$7,d0      io.sstrg
trap        #3
moveq.w     #9,d1      at#3,11,9
moveq.w     #11,d2
moveq.w     #-1,d3
move.l      id3,a0
moveq       #$10,d0     sd.pos
trap        #3
moveq.w     #14,d2      print#3,"3   Directorio"
moveq.w     #-1,d3
move.l      id3,a0
lea         mensaje3,a1
moveq       #$7,d0      io.sstrg
trap        #3
moveq.w     #9,d1       at#3,13,9
moveq.w     #13,d2
moveq.w     #-1,d3
move.l      id3,a0
moveq       #$10,d0     sd.pos
trap        #3
moveq.w     #16,d2      print#3,"4   Ver fichero"
moveq.w     #-1,d3
move.l      id3,a0
lea         mensaje4,a1
```

```
            moveq       #$7,d0      io.sstrg
            trap        #3
            moveq.w     #9,d1       at#3,16,9
            moveq.w     #16,d2
            moveq.w     #-1,d3
            move.l      id3,a0
            moveq       #$10,d0     sd.pos
            trap        #3
            moveq.w     #14,d2      print#3,"ESC  =  Salida"
            moveq.w     #-1,d3
            move.l      id3,a0
            lea         mensaje5,a1
            moveq       #$7,d0
            trap        #3
            moveq.w     #0,d1       at#3,0,0
            moveq.w     #0,d2
            moveq.w     #-1,d3
            move.l      id3,a0
            moveq       #$10,d0     sd.pos
            trap        #3
            moveq.w     #-1,d3      cursor_on#3
            move.l      id3,a0
            moveq       #$e,d0      sd.cure
            trap        #3
            moveq.w     #-1,d3      op=CODE(INKEY$(#3,-1))
            move.l      id3,a0
            moveq       #$1,d0      io.fbyte
            trap        #3
            move.b      d1,-(a7)
            moveq.w     #-1,d3      cursor_off#3
            move.l      id3,a0
            moveq       #$f,d0      sd.curs
            trap        #3
            move.b      (a7)+,d1
            cmp.b       #49,d1
            beq         qz
            cmp.b       #50,d1
            beq         zq
            cmp.b       #51,d1
            beq         unidad
            cmp.b       #52,d1
            beq         fichero
            cmp.b       #27,d1
            beq         fin
            bsr         beep
            bra         bucle
fin
            move.l      id3,a0      close#3
            moveq       #2,d0       io.close
            trap        #2
            move.l      id4,a0      close#4
            moveq       #2,d0       io.close
            trap        #2
            moveq.l     #-1,d1
            moveq.l     #0,d3
            moveq       #$5,d0      mt.frjob
            trap        #1          LLEGADO A ESTE PUNTO EL PROGRAMA HA MUERTO
* Aqui hay una serie de subrutinas que se repiten mucho a lo largo del programa
beep        lea         sonido,a3
            moveq       #$11,d0
            trap        #1
            rts
impr22
            moveq.l     #21,d4
loop22
            moveq       #32,d1
            moveq.w     #-1,d3
            move.l      id3,a0
            moveq       #$5,d0      io.sbyte
            trap        #3
```

```
            dbf         d4,loop22
            rts
input
            moveq.w     #100,d2
            moveq.w     #-1,d3
            move.l      id3,a0
            lea         buffer,a3
            lea.l       2(a3),a1
            moveq       #$2,d0      io.fline
            trap        #3
            subi.w      #1,d1
            move.w      d1,(a3)
            rts
at
            moveq.w      #11,d1     at#3,7,11
            moveq.w      #7,d2
            moveq.w      #-1,d3
            move.l       id3,a0
            moveq        #$10,d0    sd.pos
            trap         #3
            rts
bql_z88
            moveq.l     #7,d4
            moveq       #0,d5
ql_z88
            move.w      #138,d1     csize#3,138-i,30-i
            sub.w       d5,d1
            move.w      #30,d2
            sub.w       d5,d2
            moveq.w     #-1,d3
            move.l      id3,a0
            moveq       #$17,d0       sd.pixp
            trap        #3
            move.b      d5,d1
            moveq.w     #-1,d3
            move.l      id3,a0
            moveq       #$29,d0     sd.setin
            trap        #3
            moveq.w     #10,d2       print#3,"QL  >  Z88"
            moveq.w     #-1,d3
            move.l      id3,a0
            lea         men_ql1,a1
            moveq       #$7,d0
            trap        #3
            addq.w      #1,d5
            dbf         d4,ql_z88
            rts

bz88_ql
            moveq.l     #7,d4
            moveq       #0,d5
z88_ql
            move.w      #138,d1     csize#3,138-i,30-i
            sub.w       d5,d1
            move.w      #30,d2
            sub.w       d5,d2
            moveq.w     #-1,d3
            move.l      id3,a0
            moveq       #$17,d0       sd.pixp
            trap        #3
            move.b      d5,d1
            moveq.w     #-1,d3
            move.l      id3,a0
            moveq       #$29,d0     sd.setin
            trap        #3
            moveq.w     #10,d2       print#3,"Z88  >  QL"
            moveq.w     #-1,d3
            move.l      id3,a0
            lea         men_z1,a1
            moveq       #$7,d0
```

```
            trap        #3
            addq.w      #1,d5
            dbf         d4,z88_ql
            rts
* Procedimiento traductor QL a Z88
qz
            move.w      #-1,d3        cls#3
            move.l      id3,a0
            moveq       #$20,d0       sd.clear
            trap        #3
            moveq.w     #1,d1         over#3,1
            moveq.w     #-1,d3
            move.l      id3,a0
            moveq       #$2c,d0       sd.setmd
            trap        #3
            moveq.w     #2,d1         csize#3,2,1
            moveq.w     #1,d2
            moveq.w     #-1,d3
            move.l      id3,a0
            moveq       #$2d,d0       sd.setsz
            trap        #3
            bsr         bql_z88
            moveq.w     #0,d1        over#3,0
            moveq.w     #-1,d3
            move.l      id3,a0
            moveq       #$2c,d0       sd.setmd
            trap        #3
            moveq.w     #2,d1        csize#3,2,0
            moveq.w     #0,d2
            moveq.w     #-1,d3
            move.l      id3,a0
            moveq       #$2d,d0      sd.setsz
            trap        #3
            moveq       #2,d1        ink#3,2
            moveq.w     #-1,d3
            move.l      id3,a0
            moveq       #$29,d0      sd.setin
            trap        #3
            moveq.w     #3,d1        at#3,7,3
            moveq.w     #7,d2
            moveq.w     #-1,d3
            move.l      id3,a0
            moveq       #$10,d0      sd.pos
            trap        #3
            moveq.w     #8,d2       print#3,"Fuente: "
            moveq.w     #-1,d3
            move.l      id3,a0
            lea         men_ql2,a1
            moveq       #$7,d0       io.sstrg
            trap        #3
            moveq.w     #3,d1        at#3,8,3
            moveq.w     #8,d2
            moveq.w     #-1,d3
            move.l      id3,a0
            moveq       #$10,d0      sd.pos
            trap        #3
            move.w      #24,d2
            move.w      #-1,d3
            move.l      id3,a0
            lea         men_ql3,a1
            moveq       #$7,d0       io.sstrg
            trap        #3
err3
            bsr         at
            bsr         impr22
            bsr         at
            moveq.w     #-1,d3
            move.l      id3,a0
            moveq       #$e,d0       sd.cure
            trap        #3
```

```
                bsr         input
                moveq.w     #-1,d3
                move.l      id3,a0
                moveq       #$f,d0      sd.cure
                trap        #3
                moveq.l     #-1,d1
                moveq.l     #0,d3
                lea         buffer,a0
                moveq       #1,d0       io.open
                trap        #2
                tst         d0
                beq         cont_qz
                bsr         beep
                bra.s       err3
      cont_qz
                lea         id6,a3
                move.l      a0,(a3)
                move.w      #-1,d3       cls#3
                move.l      id3,a0
                moveq       #$20,d0      sd.clear
                trap        #3
                moveq.w     #1,d1        over#3,1
                moveq.w     #-1,d3
                move.l      id3,a0
                moveq       #$2c,d0      sd.setmd
                trap        #3
                moveq.w     #2,d1        csize#3,2,1
                moveq.w     #1,d2
                moveq.w     #-1,d3
                move.l      id3,a0
                moveq       #$2d,d0      sd.setsz
                trap        #3
                bsr         bql_z88
                moveq.w     #0,d1       over#3,0
                moveq.w     #-1,d3
                move.l      id3,a0
                moveq       #$2c,d0      sd.setmd
                trap        #3
                moveq.w     #2,d1       csize#3,2,0
                moveq.w     #0,d2
                moveq.w     #-1,d3
                move.l      id3,a0
                moveq       #$2d,d0     sd.setsz
                trap        #3
                moveq       #2,d1       ink#3,2
                moveq.w     #-1,d3
                move.l      id3,a0
                moveq       #$29,d0     sd.setin
                trap        #3
                moveq.w     #3,d1       at#3,7,3
                moveq.w     #7,d2
                moveq.w     #-1,d3
                move.l      id3,a0
                moveq       #$10,d0     sd.pos
                trap        #3
                moveq.w     #9,d2      print#3,"Destino: "
                moveq.w     #-1,d3
                move.l      id3,a0
                lea         men_ql4,a1
                moveq       #$7,d0      io.sstrg
                trap        #3
                moveq.w     #3,d1       at#3,8,3
                moveq.w     #8,d2
                moveq.w     #-1,d3
                move.l      id3,a0
                moveq       #$10,d0     sd.pos
                trap        #3
                move.w      #24,d2      print#3,"   (Ej.: FLP2_FichQL_TEX)"
                move.w      #-1,d3
                move.l      id3,a0
```

```
            lea         men_ql5,a1
            moveq       #$7,d0      io.sstrg
            trap        #3
err4
            bsr         at
            bsr         impr22
            bsr         at
            moveq.w     #-1,d3
            move.l      id3,a0
            moveq       #$e,d0      sd.cure
            trap        #3
            bsr         input
            moveq.w     #-1,d3
            move.l      id3,a0
            moveq       #$f,d0      sd.cure
            trap        #3
            moveq.l     #-1,d1
            moveq.l     #2,d3
            lea         buffer,a0
            moveq       #1,d0       io.open
            trap        #2
            tst         d0
            beq         cont_qz2
            bsr         beep
            bra.s       err4
cont_qz2
            lea         id5,a3
            move.l      a0,(a3)
            move.w      #-1,d3       cls#3
            move.l      id3,a0
            moveq       #$20,d0      sd.clear
            trap        #3
            moveq.w     #3,d1       at#3,12,3
            moveq.w     #12,d2
            moveq.w     #-1,d3
            move.l      id3,a0
            moveq       #$10,d0     sd.pos
            trap        #3
            moveq.w     #26,d2      print#3,"Traduciendo QL  >  Z88 ..."
            moveq.w     #-1,d3
            move.l      id3,a0
            lea         men_traz,a1
            moveq       #7,d0       io.sstrg
            trap        #3
traqz
            moveq.w     #-1,d3      a%=CODE(INKEY$(#6)
            move.l      id6,a0
            moveq       #$1,d0      io.fbyte
            trap        #3
            tst         d0
            bne         fin_fichero
            cmp.b       #10,d1
            bne         sigueq
            moveq       #13,d1
            bra         pon
sigueq
            cmp.b       #13,d1
            beq         traqz
            lea         ql,a4
            lea         z88,a5
            moveq.l     #24,d2
ascii
            cmp.b       0(a4,d2.w),d1
            beq         traq
            dbf         d2,ascii
            bra.s       pon
traq
            move.b      0(a5,d2.w),d1
pon
            moveq.w     #-1,d3
```

```
            move.l      id5,a0
            moveq       #5,d0       io.sbyte
            trap         #3
            bra         traqz
fin_fichero
            move.l      id5,a0
            moveq       #2,d0       io.close
            trap        #2
            move.l      id6,a0
            moveq       #2,d0       io.close
            trap        #2
            move.w      #-1,d3       cls#3
            move.l      id3,a0
            moveq       #$20,d0      sd.clear
            trap        #3
            moveq.w     #3,d1       at#3,16,3
            moveq.w     #16,d2
            moveq.w     #-1,d3
            move.l      id3,a0
            moveq       #$10,d0     sd.pos
            trap        #3
            moveq.w     #17,d2      print#3,"Proceso concluido"
            moveq.w     #-1,d3
            move.l      id3,a0
            lea         proceso,a1
            moveq       #$7,d0      io.sstrg
            trap        #3
            moveq.w     #25,d3      pause 25
            move.l      id3,a0
            moveq       #1,d0       io.fbyte
            trap        #3
            bra         bucle
* Traductor Z88 a QL
zq
            move.w      #-1,d3       cls#3
            move.l      id3,a0
            moveq       #$20,d0      sd.clear
            trap        #3
            moveq.w     #1,d1        over#3,1
            moveq.w     #-1,d3
            move.l      id3,a0
            moveq       #$2c,d0      sd.setmd
            trap        #3
            moveq.w     #2,d1        csize#3,2,1
            moveq.w     #1,d2
            moveq.w     #-1,d3
            move.l      id3,a0
            moveq       #$2d,d0      sd.setsz
            trap        #3
            bsr         bz88_ql
            moveq.w     #0,d1       over#3,0
            moveq.w     #-1,d3
            move.l      id3,a0
            moveq       #$2c,d0      sd.setmd
            trap        #3
            moveq.w     #2,d1       csize#3,2,0
            moveq.w     #0,d2
            moveq.w     #-1,d3
            move.l      id3,a0
            moveq       #$2d,d0     sd.setsz
            trap        #3
            moveq       #2,d1       ink#3,2
            moveq.w     #-1,d3
            move.l      id3,a0
            moveq       #$29,d0     sd.setin
            trap        #3
            moveq.w     #3,d1       at#3,7,3
            moveq.w     #7,d2
            moveq.w     #-1,d3
            move.l      id3,a0
```

```
                moveq      #$10,d0     sd.pos
                trap       #3
                moveq.w    #8,d2      print#3,"Fuente: "
                moveq.w    #-1,d3
                move.l     id3,a0
                lea        men_ql2,a1
                moveq      #$7,d0      io.sstrg
                trap       #3
                moveq.w    #3,d1       at#3,8,3
                moveq.w    #8,d2
                moveq.w    #-1,d3
                move.l     id3,a0
                moveq      #$10,d0     sd.pos
                trap       #3
                move.w     #25,d2
                move.w     #-1,d3
                move.l     id3,a0
                lea        men_z3,a1
                moveq      #$7,d0      io.sstrg
                trap       #3
      err5
                bsr        at
                bsr        impr22
                bsr        at
                moveq.w    #-1,d3
                move.l     id3,a0
                moveq      #$e,d0      sd.cure
                trap       #3
                bsr        input
                moveq.w    #-1,d3
                move.l     id3,a0
                moveq      #$f,d0      sd.cure
                trap       #3
                moveq.l    #-1,d1
                moveq.l    #0,d3
                lea        buffer,a0
                moveq      #1,d0       io.open
                trap       #2
                tst        d0
                beq        cont_zq
                bsr        beep
                bra.s      err5
      cont_zq
                lea        id6,a3
                move.l     a0,(a3)
                move.w     #-1,d3       cls#3
                move.l     id3,a0
                moveq      #$20,d0      sd.clear
                trap       #3
                moveq.w    #1,d1        over#3,1
                moveq.w    #-1,d3
                move.l     id3,a0
                moveq      #$2c,d0      sd.setmd
                trap       #3
                moveq.w    #2,d1        csize#3,2,1
                moveq.w    #1,d2
                moveq.w    #-1,d3
                move.l     id3,a0
                moveq      #$2d,d0      sd.setsz
                trap       #3
                bsr        bz88_ql
                moveq.w    #0,d1       over#3,0
                moveq.w    #-1,d3
                move.l     id3,a0
                moveq      #$2c,d0      sd.setmd
                trap       #3
                moveq.w    #2,d1       csize#3,2,0
                moveq.w    #0,d2
                moveq.w    #-1,d3
                move.l     id3,a0
```

```
                moveq       #$2d,d0     sd.setsz
                trap        #3
                moveq       #2,d1       ink#3,2
                moveq.w     #-1,d3
                move.l      id3,a0
                moveq       #$29,d0     sd.setin
                trap        #3
                moveq.w     #3,d1       at#3,7,3
                moveq.w     #7,d2
                moveq.w     #-1,d3
                move.l      id3,a0
                moveq       #$10,d0     sd.pos
                trap        #3
                moveq.w     #9,d2      print#3,"Destino: "
                moveq.w     #-1,d3
                move.l      id3,a0
                lea         men_ql4,a1
                moveq       #$7,d0      io.sstrg
                trap        #3
                moveq.w     #3,d1       at#3,8,3
                moveq.w     #8,d2
                moveq.w     #-1,d3
                move.l      id3,a0
                moveq       #$10,d0     sd.pos
                trap        #3
                move.w      #25,d2      print#3,"   (Ej.: FLP2_FichZ88_TEX)"
                move.w      #-1,d3
                move.l      id3,a0
                lea         men_z5,a1
                moveq       #$7,d0      io.sstrg
                trap        #3
     err6
                bsr         at
                bsr         impr22
                bsr         at
                moveq.w     #-1,d3
                move.l      id3,a0
                moveq       #$e,d0      sd.cure
                trap        #3
                bsr         input
                moveq.w     #-1,d3
                move.l      id3,a0
                moveq       #$f,d0      sd.cure
                trap        #3
                moveq.l     #-1,d1
                moveq.l     #2,d3
                lea         buffer,a0
                moveq       #1,d0       io.open
                trap        #2
                tst         d0
                beq         cont_zq2
                bsr         beep
                bra.s       err6
     cont_zq2
                lea         id5,a3
                move.l      a0,(a3)
                move.w      #-1,d3       cls#3
                move.l      id3,a0
                moveq       #$20,d0      sd.clear
                trap        #3
                moveq.w     #3,d1       at#3,12,3
                moveq.w     #12,d2
                moveq.w     #-1,d3
                move.l      id3,a0
                moveq       #$10,d0     sd.pos
                trap        #3
                moveq.w     #26,d2      print#3,"Traduciendo Z88  >  QL ..."
                moveq.w     #-1,d3
                move.l      id3,a0
                lea         men_traq,a1
```

```
            moveq       #7,d0       io.sstrg
            trap        #3
trazq
            moveq.w     #-1,d3      a%=CODE(INKEY$(#6)
            move.l      id6,a0
            moveq       #$1,d0      io.fbyte
            trap        #3
            tst         d0
            bne         fin_f
            cmp.b       #13,d1
            bne         siguez
            moveq       #10,d1
            bra         ponz
siguez
            lea         z88,a4
            lea         ql,a5
            moveq.l     #24,d2
asciiz
            cmp.b       0(a4,d2.w),d1
            beq         traz
            dbf         d2,asciiz
            bra.s       ponz
traz
            move.b      0(a5,d2.w),d1
ponz
            moveq.w     #-1,d3
            move.l      id5,a0
            moveq       #5,d0       io.sbyte
            trap        #3
            bra         trazq
fin_f
            move.l      id5,a0
            moveq       #2,d0       io.close
            trap        #2
            move.l      id6,a0
            moveq       #2,d0       io.close
            trap        #2
            move.w      #-1,d3       cls#3
            move.l      id3,a0
            moveq       #$20,d0      sd.clear
            trap        #3
            moveq.w     #3,d1       at#3,16,3
            moveq.w     #16,d2
            moveq.w     #-1,d3
            move.l      id3,a0
            moveq       #$10,d0     sd.pos
            trap        #3
            moveq.w     #17,d2      print#3,"Proceso concluido"
            moveq.w     #-1,d3
            move.l      id3,a0
            lea         proceso,a1
            moveq       #$7,d0      io.sstrg
            trap        #3
            moveq.w     #25,d3      pause 25
            move.l      id3,a0
            moveq       #1,d0       io.fbyte
            trap        #3
            bra         bucle
* Procedimiento directorio
unidad
err1
            moveq.w     #-1,d3      cls#3
            move.l      id3,a0
            moveq       #$20,d0     sd.clear
            trap        #3
            moveq.w     #3,d1       at#3,16,3
            moveq.w     #16,d2
            moveq.w     #-1,d3
            move.l      id3,a0
            moveq       #$10,d0     sd.pos
```

```
                trap        #3
                moveq.w     #-1,d3       cursor_on#3
                move.l      id3,a0
                moveq       #$e,d0       sd.cure
                moveq.w     #20,d2       input#3,"Unidad (Ej.: FLP1_): "
                moveq.w     #-1,d3
                move.l      id3,a0
                lea         men_u,a1
                moveq       #7,d0
                trap        #3
                moveq.w     #100,d2
                moveq.w     #-1,d3
                move.l      id3,a0
                lea         buffer,a3
                lea         2(a3),a1
                moveq       #2,d0        io.fline
                trap        #3
                subi.w      #1,d1
                move.w      d1,(a3)
                moveq.l     #-1,d1       dir#4,flp1_
                moveq.l     #4,d3
                lea         buffer,a0
                moveq       #1,d0        io.open
                trap        #2
                lea         id5,a3
                move.l      a0,(a3)
                tst.w       d0
                beq         dir
                bsr         beep
                bra.s       err1
      dir
                move.l      id5,a5           ID en A5
                move.l      id4,a4           ID SALIDA en A4
                moveq       #$45,d0
                move.l      (a6),a1          basic buffer
                bsr.s       dotrp3           obtiene detalles sobre el medio
                bne.s       excat            si error entonces cerramos y nos vamos
                move.l      d1,-(a7)         guarda sector nos
                move.l      (a6),a1          comienzo buffer
                moveq       #$0a,d2          longitud del nombre
                bsr.s       prtstr           imprime nombre
                moveq       #5,d0
                moveq       #10,d1
                trap        #3               imprime LF
                movem.w     (a7)+,d1-d2      recuperar contador de sector
                move.w      d2,d4
                bsr.s       prtnum           imprime el primer número
                moveq       #5,d0
                move.b      #'/',d1          imprime /
                trap        #3
                move.w      d4,d1
                bsr.s       prtnum           imprime el segundo número
                moveq       #5,d0
                moveq       #10,d1
                trap        #3               imprime LF
      nxname
                moveq       #3,d0
                moveq       #64,d2           número de bytes para obtener
                move.l      (a6),a1
                move.l      a5,a0            ID
                moveq       #0,d4
                bsr         dotrp3
                bne         q2               no vamos sin errores si EOF
                suba.w      #48,a1           comienzo de nombre
                move.w      -2(a6,a1.l),d2   longitud nombre
                beq.s       nxname           si nombre nulo
                bsr.s       prtstr           también escribe el nombre
                moveq       #5,d0
                moveq       #10,d1
                trap        #3               imprime LF
```

```
            bra         nxname

prtnum
            move.l      (a6),a1
            lea         2(a1),a0
            move.w      d1,0(a6,a1.l)   guarda en stack
            move.w      $f2,a2
            jsr         (a2)            convierte a ASCII
            move.w      d1,d2           longitud de número - ahora imprimelo

prtstr
            moveq       #7,d0
            move.l      a4,a0           imprime en Screen
dotrp3
            trap        #4              hazlo absoluto
            moveq       #-1,d3          infinito timeout
            trap        #3
            tst.l       d0              test para error
            rts
excat       move.l      d0,d4           cierra ID
q2          moveq       #2,d0
            trap        #2
            move.l      d4,d0
            move.w      #-1,d3      pause
            move.l      id3,a0
            moveq       #$1,d0      io.fbyte
            trap        #3
            bra         bucle
* Procedimiento ver un fichero en pantalla
fichero
err2
            moveq.w     #-1,d3      cls#3
            move.l      id3,a0
            moveq       #$20,d0     sd.clear
            trap        #3
            moveq.w     #3,d1       at#3,16,3
            moveq.w     #16,d2
            moveq.w     #-1,d3
            move.l      id3,a0
            moveq       #$10,d0     sd.pos
            trap        #3
            moveq.w     #-1,d3      cursor_on#3
            move.l      id3,a0
            moveq       #$e,d0      sd.cure
            moveq.w     #9,d2      input#3,"Fichero: "
            moveq.w     #-1,d3
            move.l      id3,a0
            lea         men_fich,a1
            moveq       #7,d0
            trap        #3
            moveq.w     #100,d2
            moveq.w     #-1,d3
            move.l      id3,a0
            lea         buffer,a3
            lea         2(a3),a1
            moveq       #2,d0       io.fline
            trap        #3
            subi.w      #1,d1
            move.w      d1,(a3)
            moveq.l     #-1,d1      view#4,flp1_fichero
            moveq.l     #0,d3
            lea         buffer,a0
            moveq       #1,d0       io.open
            trap        #2
            lea         id5,a3
            move.l      a0,(a3)
            tst.w       d0
            beq         dir2
            bsr         beep
            bra.s       err2
```

```
dir2
            moveq.w     #-1,d3
            move.l      id5,a0
            moveq       #1,d0       io.fbyte
            trap        #3
            tst         d0
            bne         fin_fich
            moveq.w     #-1,d3
            move.l      id4,a0
            moveq       #5,d0       io.sbyte
            trap        #3
            bra.s       dir2
fin_fich
            move.l      id5,a0
            moveq       #2,d0
            trap        #2          io.close
            bra         bucle



sonido
            dc.b        $0a,8
            dc.l        $0000aaaa
            dc.b        200,0
            dc.b        0,0,30,0
            dc.b        0,0
            dc.b        1
            cnop        0,4
id3         ds.l        1
id4         ds.l        1
id5         ds.l        1
id6         ds.l        1
buffer      ds.w        1
            ds.b        100
men_traq    dc.b        'Traduciendo Z88  ?  QL ...'
men_traz    dc.b        'Traduciendo QL  ?  Z88 ...'
men_fich    dc.b        'Fichero:  '
men_u       dc.b        'Unidad (Ej.: FLP1_): '
proceso     dc.b        'Proceso concluido '
men_z1      dc.b        'Z88  ?  QL'
men_ql1     dc.b        'QL  ?  Z88'
men_ql2     dc.b        'Fuente: '
men_ql3     dc.b        '  (Ej.: FLP1_FichQL_TEX)'
men_z3      dc.b        '  (Ej.: FLP1_FichZ88_TEX) '
men_ql4     dc.b        'Destino:  '
men_ql5     dc.b        '   (Ej.: FLP2_FichQL_TEX)'
men_z5      dc.b        '   (Ej.: FLP2_FichZ88_TEX) '
mensaje1    dc.b        '1   QL  ?  Z88'
mensaje2    dc.b        '2   Z88 ?   QL'
mensaje3    dc.b        '3   Directorio'
mensaje4    dc.b        '4   Ver fichero '
mensaje5    dc.b        'ESC  =  Salida'
mensaje     dc.b        'TRADUCTOR '
con3        dc.w        17
            dc.b        'con_400x200a56x23 '
scr4        dc.w        17
            dc.b        'scr_360x180a76x33 '
ql
            dc.b        179,96,180,168
            dc.b        169,141,140,130
            dc.b        136,144,131,145
            dc.b        148,147,149,146
            dc.b        137,151,150,152
            dc.b        132,154,153,155
            dc.b        135,0
z88
            dc.b        161,163,191,199
            dc.b        209,224,225,226
            dc.b        231,232,233,234
            dc.b        236,237,238,239
```

```
          dc.b       241,242,243,244
          dc.b       246,249,250,251
          dc.b       252
          end
```

El programita solamente ocupa 2860 bytes. Si no hubiese actualizado innecesariamente tantas veces D3 y A0, habria ocupado menos bytes.

La velocidad no está nada mal, es 7 veces más rápido que la versión Turbo-compilada. Pero aún podemos hacerlo más rápido leyendo más bytes a la vez, en vez de uno a uno (la versión más rápida seria cargar todo el fichero en memoria, hacer alli todas las traducciones y luego grabarlo entero de golpe).

En la rutina "ver fichero" no estaria nada mal leer el teclado durante unos milisegundos para comprobar si se ha pulsado una tecla encargada de abortar, pues ver un fichero entero muy grande podria ser fatigoso (esta subrutina se apropia de la pantalla para ella solita. No le gusta compartir. Es muy avariciosa).

S. Merino, 17/10/1989



Z88

¿ES NECESARIO EL PROGRAMA TRADUCTOR QL/Z88?

Quizás si, quizás no, pero se me ocurrio hacerlo y eso es todo lo que se me ocurre decir. Realmente en un QL no era tan necesario, porque si nos acordamos de nuestras viejas técnicas, pero fiables. Nos encontramos ante un simple COPY SER2_ TO FLP2_FICHERO (cuando termina la transmisión, hay que hacer un BREAK en el QL).
¿Qué significa eso?
Muy simple, si solamente hubiesemos hecho en el programa PrinterED las traducciones de los caracteres Españoles por los correspondientes códigos del QL, habriamos tenido un fichero traducido totalmente y listo para Exportar en Turbo_Xchange v3.87. Lástima que esa opción solamente sirve cuando usamos la opción imprimir en impresora.
Y ¿Porque no hemos usado ese método? Simplemente, no nos dió la gana y no es muy profesional (aunque es tan bueno y más rápido que el anterior).

MARCOS ME HA CRITICADO MI CHAPUZA EN VERANO

Realmente tiene toda la razón, no he respetado los margenes ni justificante a la derecha.
Motivos: una pereza gigantesca (como todos).
En Pipedream es standard 73 caracteres por linea y un mapa de página en pantalla, pero si quitamos el mapa, podemos ampliar 2 nuevas columnas (ver EDIT opción add column), y pulsando la tecla que tiene un cuadrado dibujado y la flecha derecha del cursor 7 veces podemos ampliar las lineas a 80 caracteres visibles en pantalla (naturalmente todo ello perdiendo el maravilloso mapa).
Lo que si he observado es que por más que uso el comando justify del QUILL, no hay manera de arreglar el margen derecho. Y el margen izquierdo se mueve a la altura del Indent. Naturalmente, quizás solamente ocurre con el sistema QZ Transfer y Traductor.
De todas formas, existe un código en la impresora que nos soluciona ese detalle del margen derecho.

NO TENEMOS CABLE DE IMPRESORA

¡Qué no cunda el pánico! Con un simple COPY SER2 TO SER1 (PULSAR BREAK FINAL TRANSMISION), es suficiente.
Es una sorprendente chapuza, pero el precio de un cable PARALELO no es de broma, y el de serie, más barato, siempre cuesta varios billetes verdes.
Aunque después de todo, en el programa QZ TRANSFER hay una opción para esto.

S. Merino, 22/10/1989.